Esercizi Calcolabilità e Complessità

Esercizio 1

Siano L1 e L2 in NP, si costruisca la TM T che decide L1 ∩ L2. L(T) è in NP?

Esercizio 2

Si dimostri che il linguaggio

$$L = \{ \langle M \rangle \mid M \text{ è una TM e } L(M) = 1^*\}$$

è indecidibile, definendo una (mapping) riduzione da A_{TM}.

L è Turing riconoscibile?

Consideriamo l'alfabeto binario come alfabeto di input.

Esercizio 2 soluzione

Dimostriamo che $A_{TM} \leq L = \{$ <M> | M è una TM e $L(M) = 1^*\}$.
La riduzione deve mandare un'istanza di A_{TM} , <M,w>, in un'istanza di L, T, in modo tale che <M,w> è in A_{TM} sse T è in L.
La riduzione è calcolata dalla seguente TM R:
R=
input <M,w>
output: la codifica della TM T di seguito descritta.

```
T =
input x
if x non è in 1*, then rifiuta
else
    esegui M su w
    if M accetta w then accetta
    if  M rifiuta then rifiuta
```

Se M accetta w allora $L(T) = 1^*$ altrimenti, cioè sia che M rifiuti w che nel caso che M non si fermi su w, $L(T) = \varnothing$ e quindi la riduzione è corretta. In virtù della riduzione L è indecidibile.

Esercizio 2 soluzione

Per la Turing riconoscibilità dovremmo costruire una TM che riconosce L.
Osserviamo che la riduzione $A_{TM} \leq L$ già mi dice che complemento di L non è Turing riconoscibile, perchè equivale a $\neg A_{TM} \leq \neg L$ e sappiamo che $\neg A_{TM}$ non è Turing riconoscibile. Se si riduce $A_{TM} \leq \neg L$, si dimostra che anche L non è Turing riconoscibile, perchè equivale a dimostrare $\neg A_{TM} \leq L$.

R= input <M,w>
    output: la codifica della TM T di seguito descritta.
    T = input x
    if x è in 1* then accetta
    else
        esegui M su w
        if M accetta w then T accetta x
        if M rifiuta w then T rifiuta x

Se M accetta w allora L(T) = {0,1}*, mentre se M non accetta w, perchè la rifiuta o perchè non si ferma su w, allora L(T) = 1*.
Quindi a un'istanza sì di A_{TM} si è associata un'istanza sì di $\neg L$, e a un'istanza no di ATM un'istanza no di $\neg L$, cioè una TM T in L.

Esercizio 3

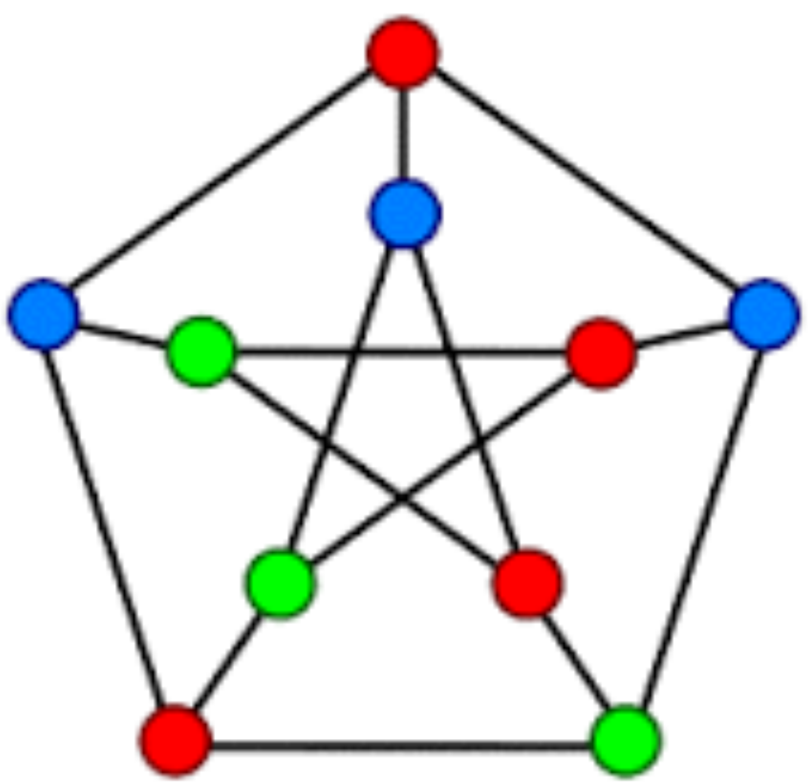
Se L1 e L2 sono in NP, è vero che L1 - L2 è in NP?

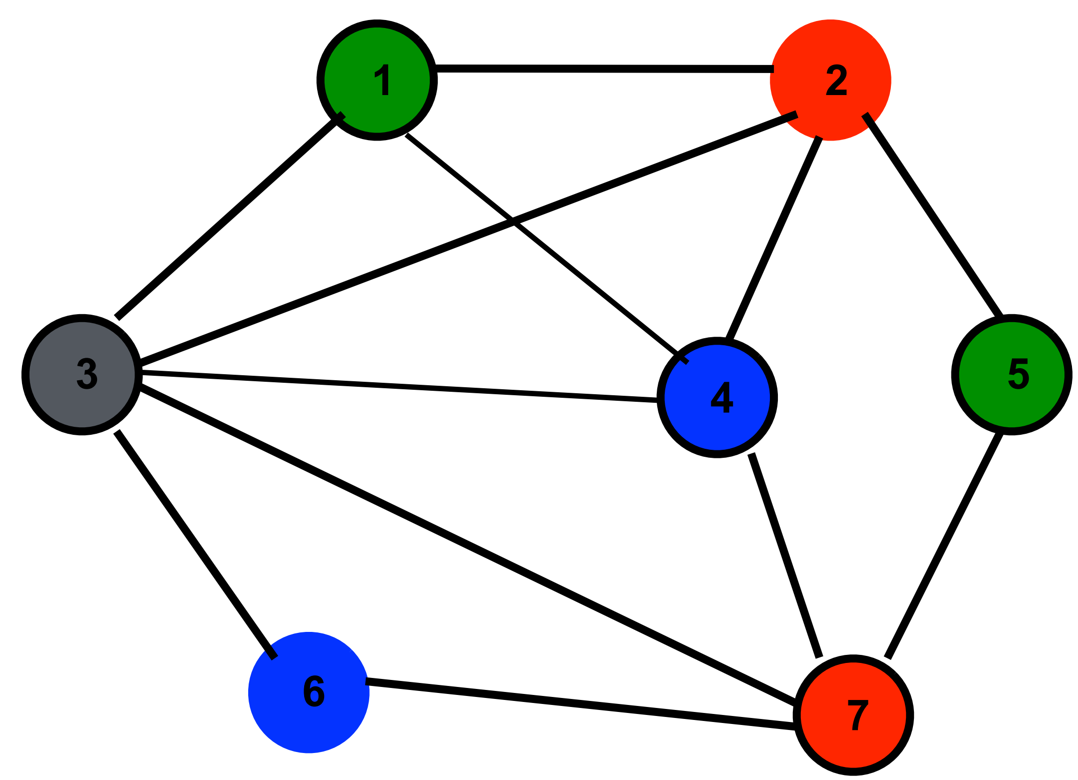
Esercizio 4

Si consideri il seguente problema:

4-COLORING ={ <G> | G è un grafo non diretto e colorabile con 4 colori}

Si dimostri che 3-COLORING si riduce polinomialmente a 4-COLORING.
É vero che 4-coloring è NP-completo?





Esercizio 5

Un linguaggio B è PSPACE-hard se $A \leq_P B$ per ogni A in PSPACE.

Si dimostri che se un linguaggio L PSPACE-hard è in NP allora

PSPACE = NP.

Esercizio 6

Si dimostri che il problema della fermata totale per TM è indecidibile.
Il problema della fermata totale è il problema di stabilire se una TM T si ferma su tutti gli input:

$$\mathrm{HALTtot}_{TM}=\{\langle T\rangle \mid \text{per ogni input } x \text{ } T \text{ si ferma su } x\}.$$

Esercizio 6 soluzione

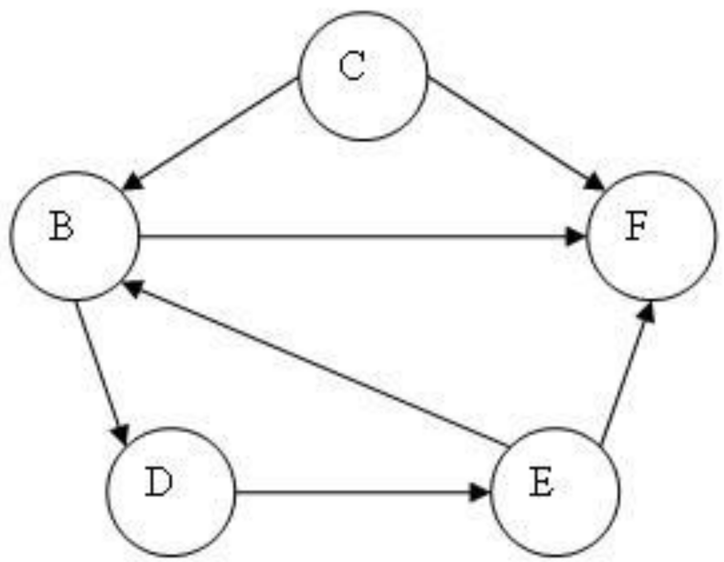
Dimostriamo che $A_{TM} \leq HALTtot_{TM}$={<T> | per ogni input x T si ferma su x}.

La riduzione deve mandare un'istanza di A_{TM} , <M,w>, in un'istanza di $HALTtot_{TM}$, T, in modo tale che <M,w> è in A_{TM} sse T è in $HALTtot_{TM}$.
La riduzione è calcolata dalla seguente TM R:
R=
input <M,w>
output: la codifica della TM T di seguito descritta.
T =
input x
1. esegui M su w
2. if M accetta w then accetta x
3. if M rifiuta w then
4. torna al punto 1.

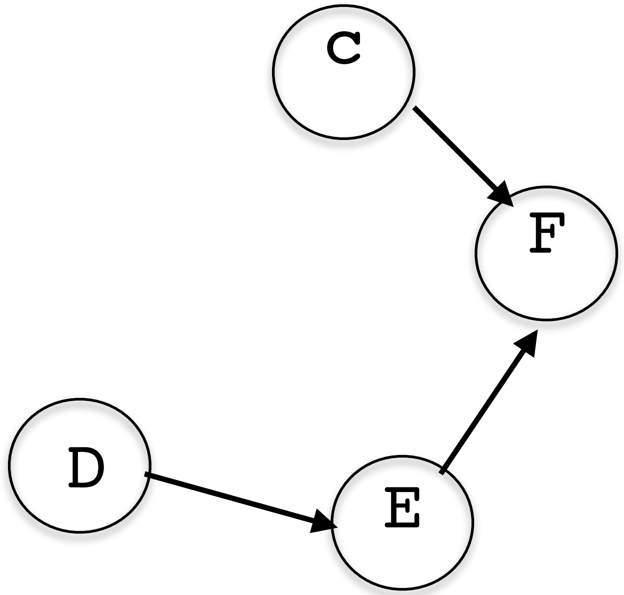
Se M accetta w allora L(T) = {0,1}* e quindi T si ferma su ogni input.
Se M rifiuta w allora L(T) = ∅, e T non si ferma su ogni input.
Se M non si ferma su w anche T non si ferma su ogni input.
Quindi <M,w> è in A_{TM} sse T è in $HALTtot_{TM}$
In virtù della riduzione L è indecidibile.

Esercizio 7 - definizioni ed esempi

Dato un grafo diretto G=(V,E), chiamiamo insieme di ritorno un sottoinsieme proprio X di vertici con la proprietà che il sottografo ottenuto dalla rimozione di X in G, G/X, è senza cicli.
G/X è il grafo ottenuto eliminando da V i vertici in X e da E gli archi incidenti i vertici di X.
Esempio di grafo in cui c'è un insieme di ritorno di 1 elemento e in cui anche ogni sottoinsieme che contiene quell'elemento è un insieme di ritorno:

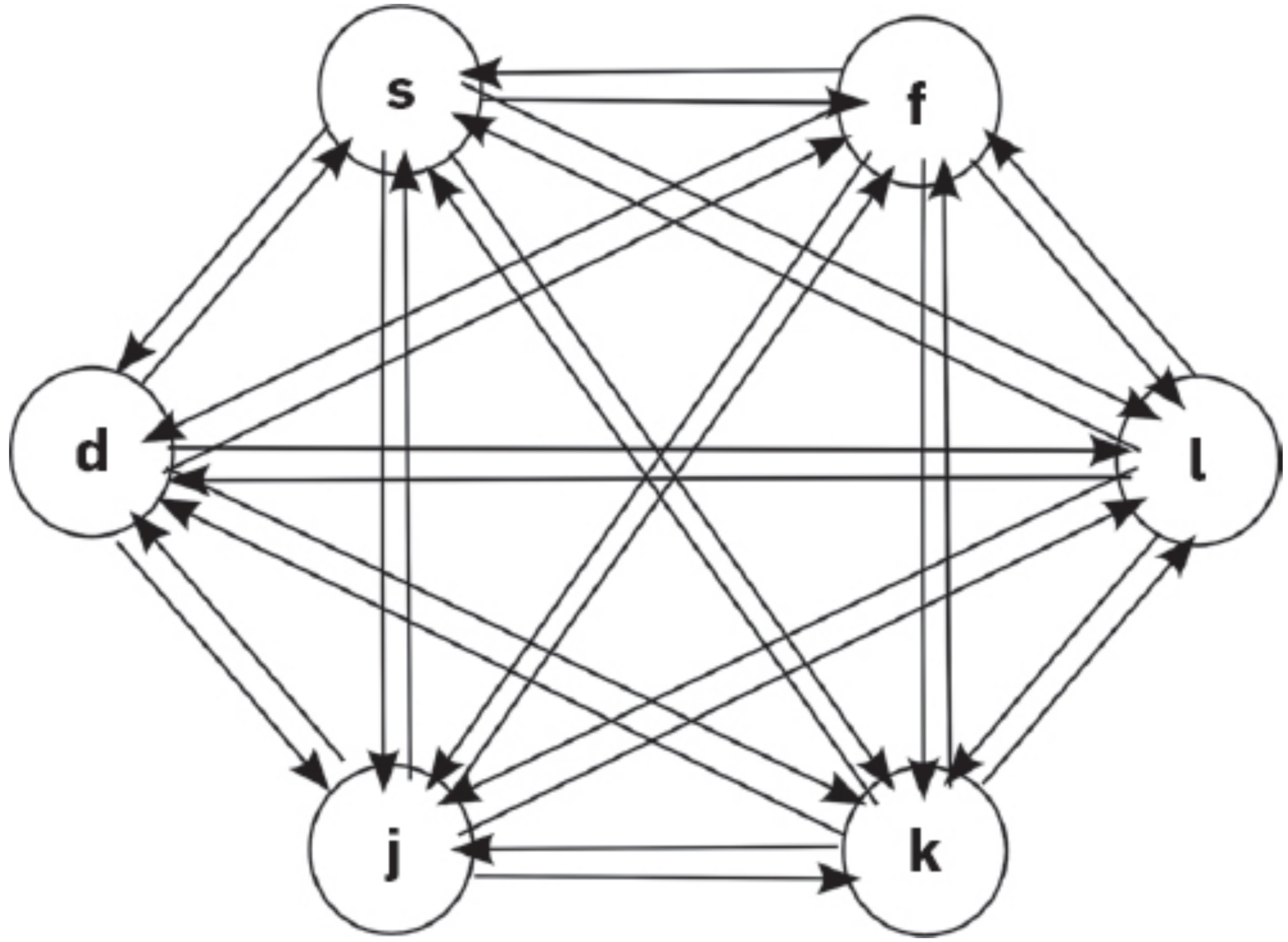


X = {B} allora G/X =



Esercizio 7

Dato un grafo diretto G=(V,E), chiamiamo insieme di ritorno un sottoinsieme proprio X di vertici con la proprietà che il sottografo ottenuto dalla rimozione di X in G, G/X, è senza cicli.
G/X è il grafo ottenuto eliminando da V i vertici in X e da E gli archi incidenti i vertici di X.
Esempio di grafo per il quale una qualunque scelta di 5 vertici fornisce un insieme di ritorno, ma in cui ogni scelta di meno di 5 vertici non va bene:



Esercizio 7

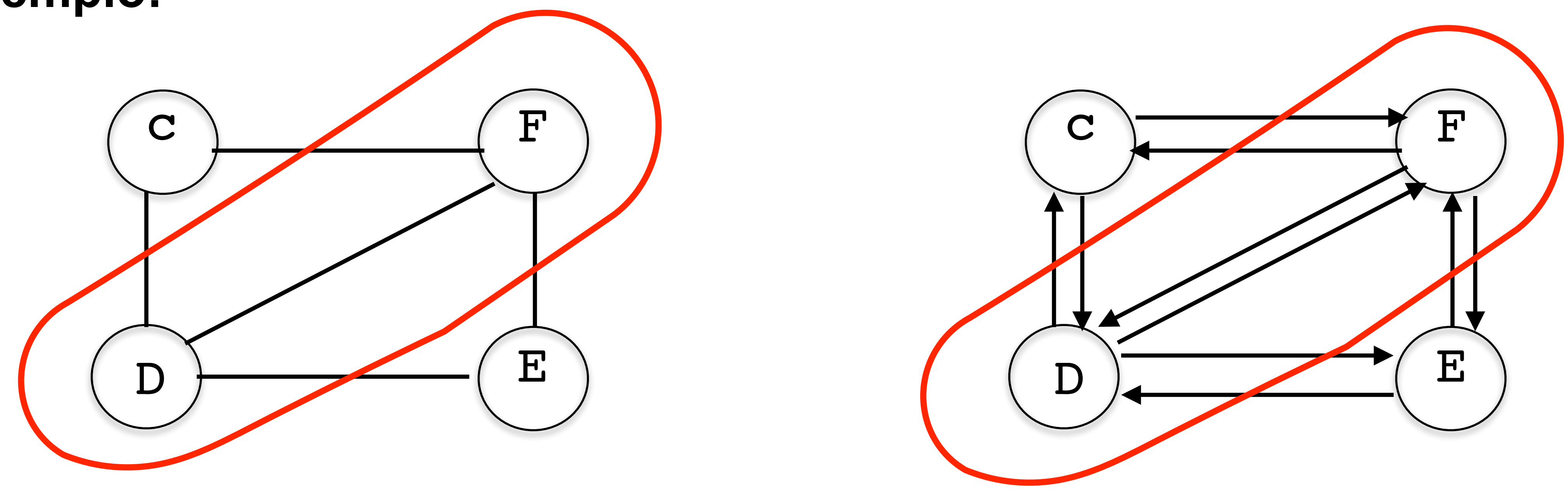
Dato un grafo diretto G=(V,E), chiamiamo insieme di ritorno un sottoinsieme proprio X di vertici con la proprietà che il sottografo ottenuto dalla rimozione di X in G, G/X, è senza cicli.
G/X è il grafo ottenuto eliminando da V i vertici in X e da E gli archi incidenti i vertici di X.
Consideriamo il problema RIT di stabilire, dato un grafo diretto G e un intero k, se G ha un insieme di ritorno di k vertici.
Si dimostri che il problema è NP-completo.

Si tratta di dimostrare che il problema è in NP e che è NP-hard, cioè che ogni problema in NP si riduce polinomialmente ad esso.

1. Il problema è in NP, infatti un verificatore polinomiale V per RIT è facilmente definito. V prende in input un'istanza del problema, cioè un grafo diretto G=(V,E) e un intero positivo k e un certificato costituito da un sottoinsieme X di vertici di V.
V verifica che X contenga k vertici, genera G/X e ne verifica l'aciclicità.
Questi controlli sono tutti eseguibili in tempo polinomiale.

Esercizio 7 soluzione

2. Il problema si può ottenere per riduzione da Vertex Cover.
Sia G = (V,E) un grafo non diretto e k un intero.
Associamo a G un grafo diretto G' = (V',E') con
V' = V e
E'= {(u,v), (v,u) | {u,v} è un arco in G}
Esempio:



Faremo vedere che un sottoinsieme X di vertici di G che è un Vertex Cover determina un insieme di ritorno di G' e viceversa.
Nell'esempio prendiamo il 2-vertex-cover X = {D,F}, se eliminiamo D ed F da G' otteniamo un grafo di due vertici sconnessi, quindi certamente un grafo aciclico.
D'altra parte se eliminando X in G' si ottiene un grafo aciclico allora vuol dire che il grafo risultante è sconnesso e quindi che ogni arco di G incideva D o F e quindi che X è un vertex cover per G.

Esercizio 7 soluzione

2.Si può definire una riduzione da Vertex Cover.
Sia G = (V,E) un grafo non diretto e k un intero.
Associamo a G un grafo diretto G' = (V',E') con
V' = V e
E'= {(u,v), (v,u) | {u,v} è un arco in G}

X è un sottoinsieme di vertici di G che è un Vertex Cover sse X è un insieme di ritorno di G.
Dimostriamo che se X è un Vertex Cover allora X è in insieme di ritorno di G'.
Se prendiamo un arco {u,v} di G, sappiamo che u o v deve essere in X, supponiamo sia u, l'altro caso essendo simmetrico. Allora l'arco (u,v) e l'arco (v,u) sono eliminati da G'/X, che quindi è un grafo senza archi e come tale aciclico.

Sia ora X un insieme di ritorno di G', allora ogni ciclo è spezzato da X, in particolare i cicli di lunghezza 2 in C={(u,v), (v,u) | u,v sono in V}, ma allora X è un Vertex Cover per G perchè ogni arco ha un vertice in X.

Esercizio 8: SET-packing

Un **SET-packing** è una famiglia di sottoinsiemi di un insieme a due a due disgiunti

Problema: Dato un insieme U, una famiglia di suoi sottoinsiemi $S_1,\ldots,S_n$ e un intero k, vogliamo sapere se c'è una famiglia C di k < n sottoinsiemi tali che $S_i \cap S_j = \emptyset$, per ogni i e j tali che S_i, S_j e in C

.

Esempio:

U = 1,2,3,4 5,6,7,8 9,10

k = 3

S_1 = 1,2

S_2 = 2,3, 4,5

S_3 = 4,7

S_4 = 5,8

S_5 = 6,7, 8,10

S_7 = 9,10

S_6 = 1,3 6,9

$\{S_1,S_3,S_4\}$ è un 3-set-packing.

Esercizio 8: set packing è NP-completo

Set packing: dato un insieme U, una famiglia di sottoinsiemi di U, $S_1,\ldots,S_m$, con m>1, e un intero positivo k, c'è un insieme $C \subseteq \{S_1,\ldots,S_m\}$ di almeno k elementi tali che $S_i \cap S_j = \varnothing$ per ogni $S_i, S_j \in C, 1 \le i \ne j \le m$?
Si dimostri che il problema è NP-completo.

Esercizio 9

Si dimostri che il seguente linguaggio L è indecidibile, costruendo una (mapping) riduzione da A_{TM}. Consideriamo l'alfabeto binario come alfabeto di input.

L = { <M> | M è una TM e accetta solo 010}.

L è Turing riconoscibile?